CMPS 3390 Lab 3

Fall 2024

Part 1 - Setup

- 1. We will be using IntelliJ Idea Community for this lab.
 - You can download and install this from Jetbrains Website.
 - If you will be completing this lab on the class/lab computers, run this command from the machine's terminal to install the JetBrains ToolBox (NOT ON ODIN!):

bash <(curl -s https://cs.csub.edu/~paul/cs3390/scripts/installJTB.sh)</pre>

Then use the ToolBox to install IDEA Community

2. Set up a new java project and make sure you use the IntelliJ Build System and openjdk-22:

New Project		×				
Q	Name:	StudentApp				
New Project	Nume.					
🖵 Java	Location:	~\IdeaProjects				
🖸 Kotlin	Project will be created in: ~\ldeaProjects\StudentApp					
G Groovy						
Co Empty Project						
Generators	Build system:	IntelliJ Maven Gradle				
<i>M</i> Maven Archetype	JDK:	G openjdk-22 Oracle OpenJDK 22.0.2 ∽				
🗇 JavaFX						
🖒 Spring 🔂	Add sample code					
Compose for Desktop						
	> Advanced Settings					
More via plugins						
?		Create				

Part 2 - Guided Java GUI Development

For our UI we will be using intelliJ's Swing UI Designer, which will allow us to drag and drop UI elements into Panel/View. Create one by clicking your source folder and selecting New > Swing UI Designer > GUI Form. Name it "StudentGUI". This will create two associate files: **StudentGUI.java** and **StudentGUI.form**

<u>N</u> ew >	© Java Class	
% Cut Ctrl+X [®] <u>C</u> opy Ctrl+C Copy Path/Reference [®] <u>P</u> aste Ctrl+V	☑ Kotlin Class/File ➡ File ■ Scratch File Ctrl+Alt+Shift+Insert Package □ Package	
Find Usages Alt+F7	⊈ package-info.java ⊈ module-info.java	
Find in Files Ctrl+Shift+F Repl <u>a</u> ce in Files Ctrl+Shift+R Analyze >	 HTML File Resource Bundle EditorConfig File 	
<u>R</u> efactor >	Swing UI Designer	🗇 GUI Form
Bookmarks		🗖 Create Dialog Class

Create the following UI structure in StudentGUI.form:

- A JPanel with the **field name** set to <u>panel</u> and the **Layout Manager** set to <u>BorderLayout</u>:
 - A JScrollPane with the **Border Side** set to <u>Center</u>
 - A JList with the **field name** set to <u>studentList</u>
 - A JPanel with the Layout Manager set to GridBagLayout and the Border Side set to East
 - A JLabel (labelFirstName) and a JTextField (inputFirstName) for first name.
 - A JLabel (labelLastName) and a JTextField (inputLastName) for last name.
 - A JLabel (labelStudentID) and a JTextField (inputStudentID) for student ID.
 - A JPanel with the **Layout Manager** set to <u>FlowLayout</u>
 - A JButton (addStudentButton) to add a student
 - A JButton (removeStudentButton) to remove a student

🎯 StudentApp.java	© StudentGUI.java	☐ StudentGUI.form →	<		
Component Tree		@ —			
🗸 🤀 Form (StudentGUI			Student List		иун 888
∽ 🛛 panel : JPanel			studentList:JList	First Nam	ne
✓ □ "Student List					
i≣ studentLi	ist : JList			Last Nam	ne
✓ □ JPanel				Student I	D
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	tName : JLabel				
	tName : JTextField				
	Name : JLabel				
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	tName : JTextField				
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	lentID : JLabel				
	dentID : JTextField				
Vertical S					
⊡ Vertical S ✓ □ JPanel	pacer				
	entButton : JButton				
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	tudentButton : JButton		B		
			• Ad	d Student Ren	nove Student

To create listeners for the JButtons, right click on them and select "Create Listener" then select Action Listener. This will automatically generate the function definition for the corresponding button in **StudentGUI.java**

Part 3 - Get It Working

Now that the basic UI is set up, and we have listeners attach to both buttons, complete the following:

- Get the panel to show up when you launch the program
- Make sure the process ends when you close the app
- When the user clicks the Add Student button:
 - Get the string values from all three input fields
 - Make sure they are not blank
 - Format the values into one string
 - Add the string to the JList (studentList)
 - Clear all three input fields
- When the user clicks the Remove Student button:
 - Make sure one of the items from the JList is selected
 - Get the index of the selected item
 - Remove the selected item from the list